**Open Ended Lab -1**

**Spring 2025**

**Course Title: Data Structures Lab**

**Course Code:** **CSE1302**

**Section: 02**

**Submitted by:**

**Student Name: Sakib Al Hasan**

**ID: 241014134**

**Department of CSE**

**University of Liberal Arts Bangladesh (ULAB)**

## **Console-Based CRUD Application**

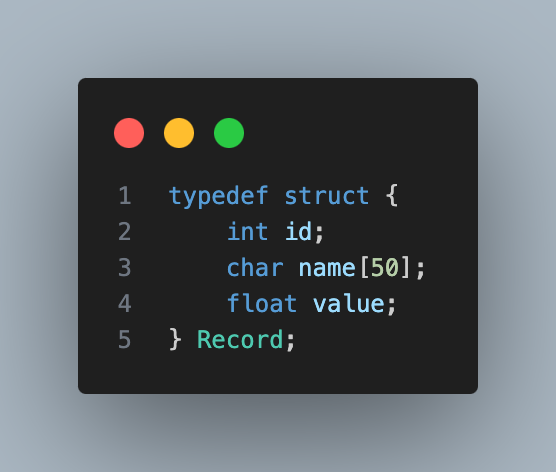
## **Objective:**

The purpose of this experiment is to design and implement a simple **console-based CRUD (Create, Read, Update, Delete)** application using the C programming language. The program manages a list of records, each containing a unique ID, a name, and a numeric value. The system utilizes **dynamic memory allocation** to handle records efficiently during runtime.

## **Explanation of Data Structures Used**

### **Data Structure:**

The system uses a **dynamically allocated array of structures** to store all records. Each record is represented using the following structure:



### **Dynamic Memory Allocation:**

* Records are stored using malloc() and realloc() to dynamically allocate memory as new records are added.
* free() is used to release memory after records are deleted or when the program exits.

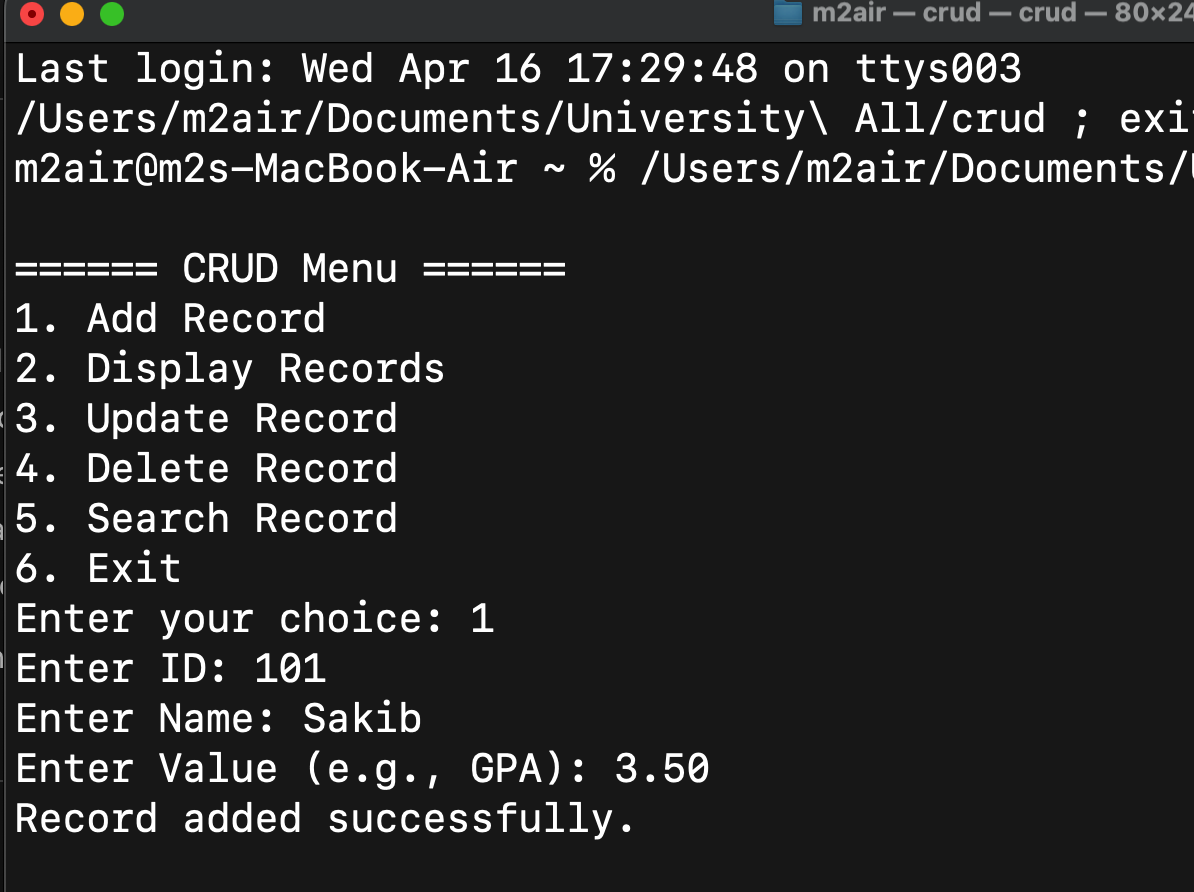
This approach allows the program to manage memory efficiently and scale with user input during execution.

## **Full CRUD Implementation and Sample Output :**

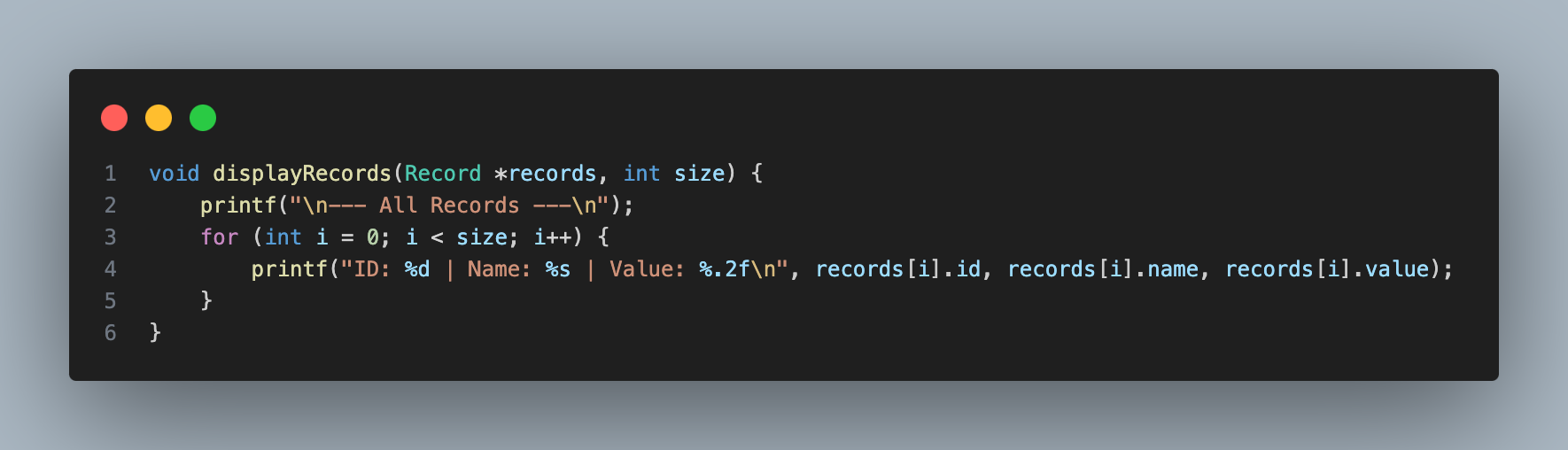
### **Create (Add a Record):**



Output:



### **Read (Display All Records):**



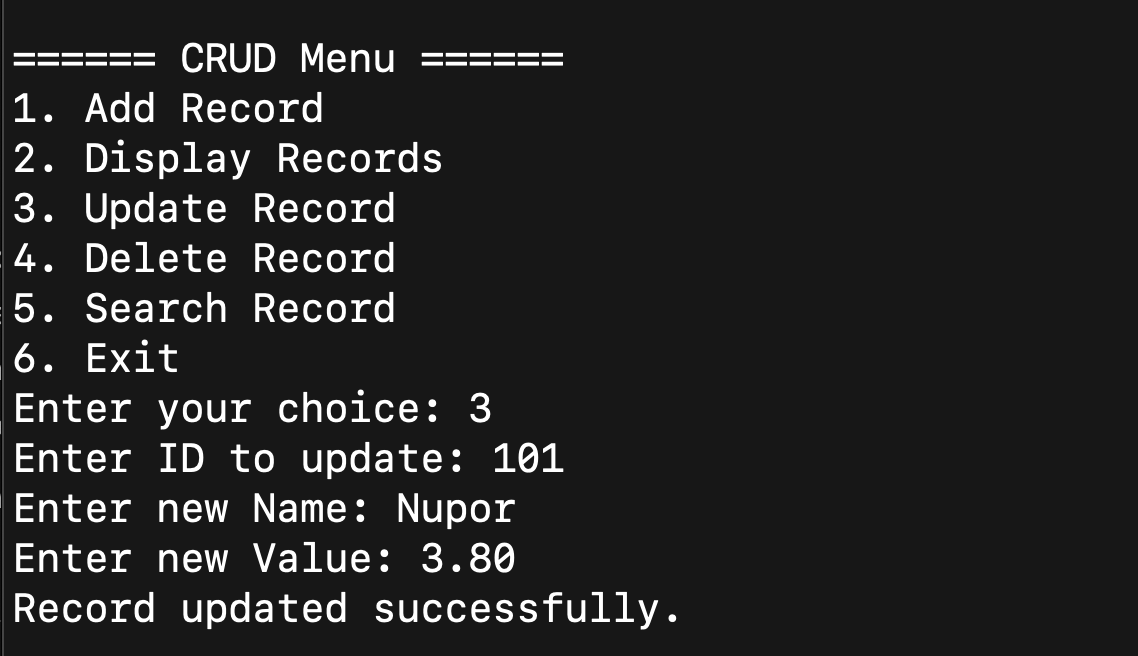
Output:



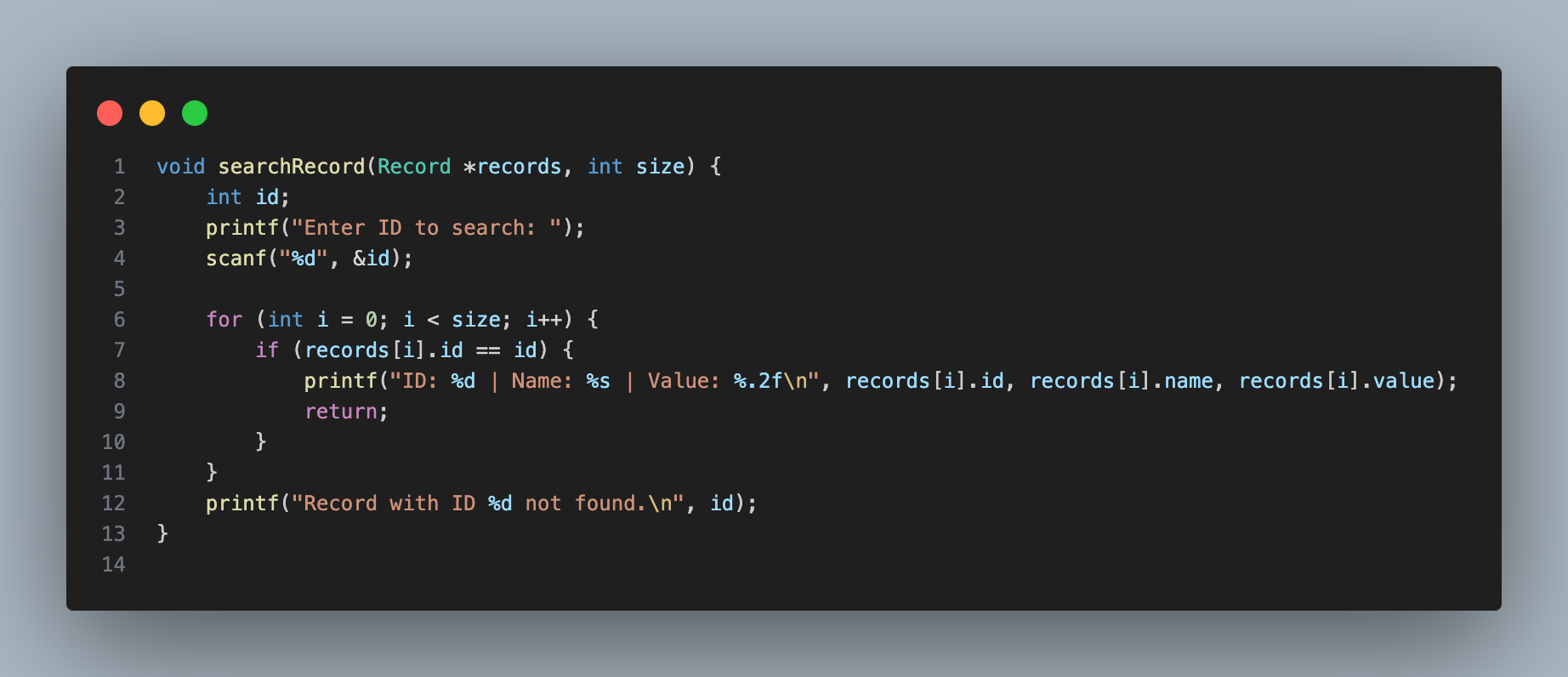
### **Update (Modify a Record by ID):**



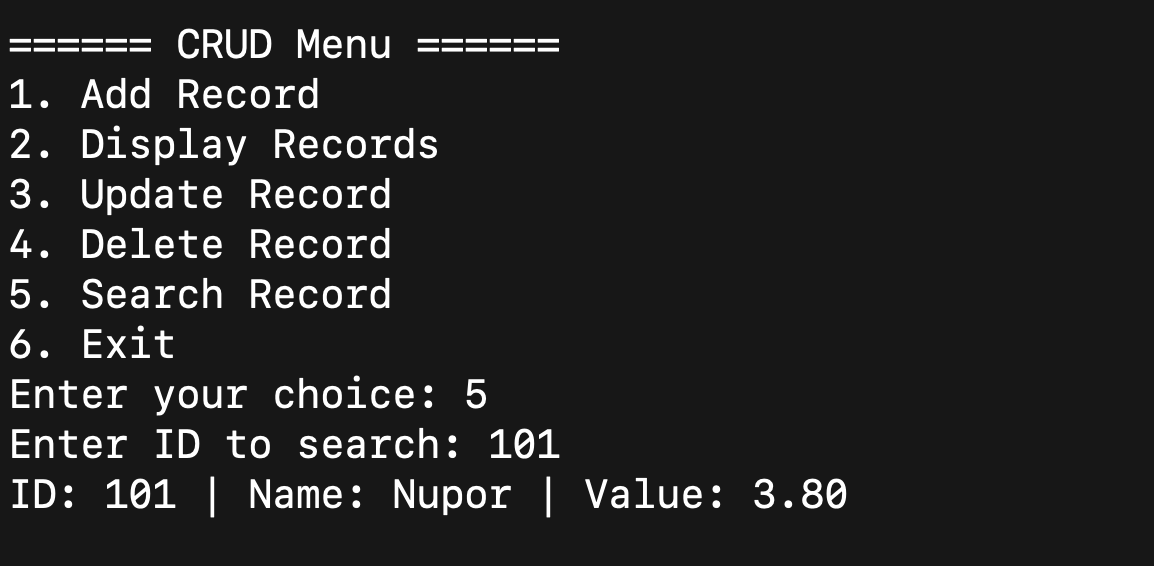
Output:



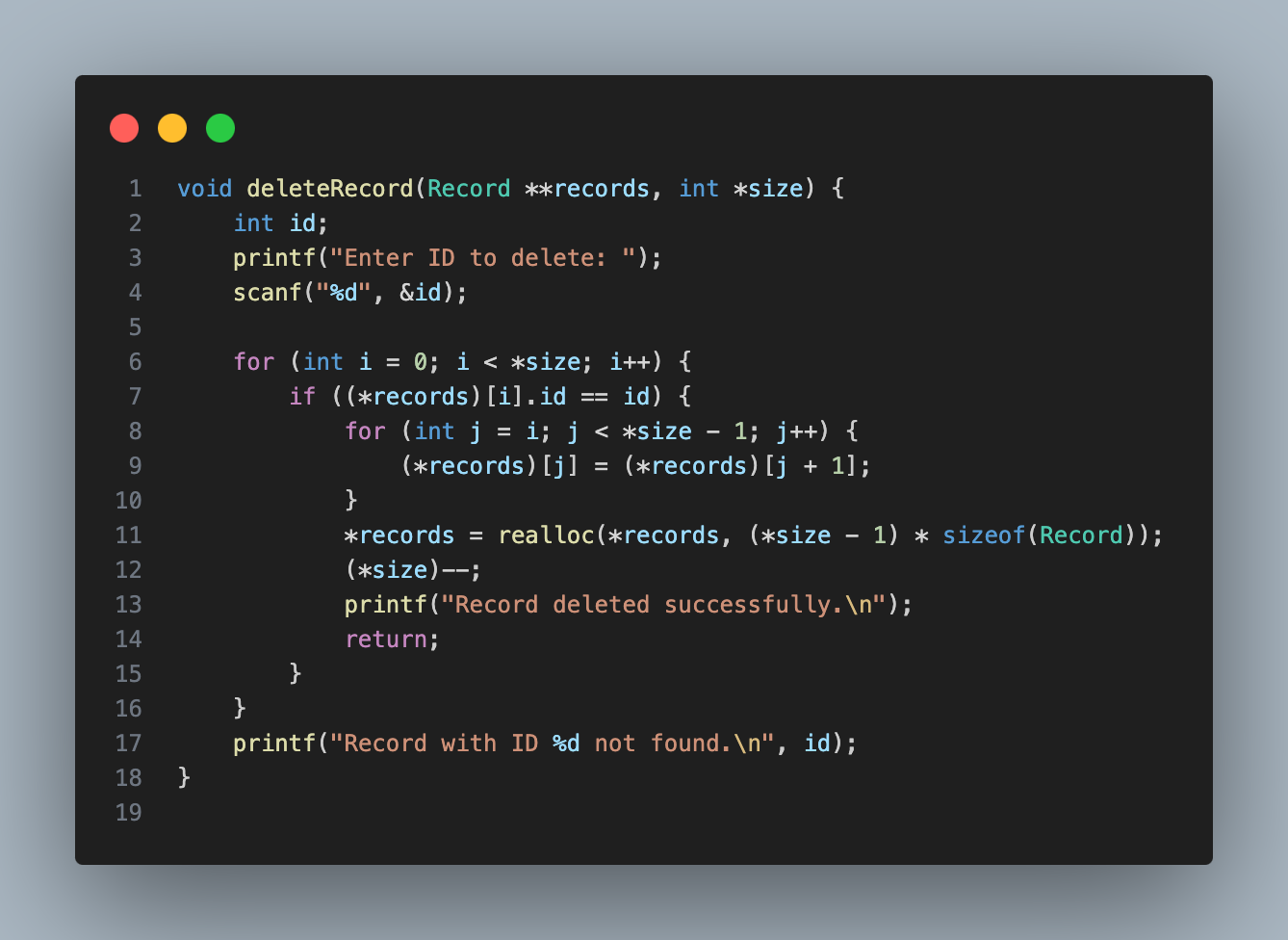
### **Search (Find a Record by ID):**



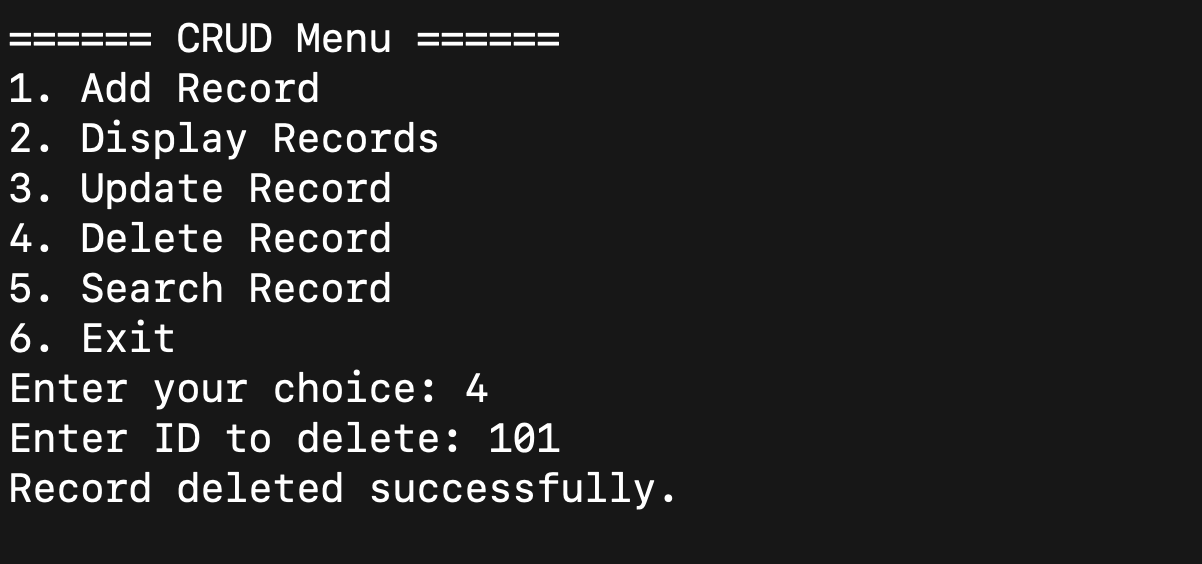
Output:



### **Delete (Remove a Record by ID):**



Output:



## **Sample Main Menu**



## **Lab Report Summary**

This project demonstrates the use of **dynamic memory allocation** and **arrays of structures** to manage data in real-time. All basic CRUD operations were implemented and tested through a user-friendly, menu-driven interface.

### Key Features:

* Dynamic memory allocation using malloc() and realloc()
* Clean modular functions for each operation
* Safe and interactive record management

### **Conclusion**

The Console-Based CRUD Application successfully illustrates how core programming concepts like **structures**, **dynamic memory**, and **modular design** can be applied to solve real-world problems. This project improves understanding of memory management, user input handling, and data processing in C.